

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID   : ZRHM-PK-05-JP;
%put NOTE: Program Name        : t_bmi.sas;
%put NOTE: Purpose              : table of summary of weight and bmi;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADSL ADAM.ADVS;
%put NOTE: Output               : t_15_2_6_16(bmi);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_aobyrne;
%put NOTE: Creation Date        : 2014-08-06;
%put NOTE: SAS Version          : 9.3;
%put NOTE: ;
%put NOTE: == Latest Run
=====;
%put NOTE: Run by                : &sysuserid;
%put NOTE: Date/Time             :
%sysfunc(putn(%sysfunc(date()),e8601da.))T%sysfunc(putn(%sysfunc(time()),
e86011z.));
%put NOTE: ;
%put NOTE: == Modification History
=====;
%put NOTE: Date      Initials   No. Reason;
%put NOTE: 12Aug2014  AOB        1) Amended Obese category;
%put NOTE: 12Aug2014  AOB        2) Treatment headers amended;
%put NOTE: 12Aug2014  AOB        3) Dp of n amended;
%put NOTE: 12Aug2014  AOB        4) Column widths amended;
%put NOTE: 12Aug2014  JMH        5) Amended wrapping of randomized in
column header;
%put NOTE: 23Sep2014  JMH        6) Added BMI classification for Day 4;
%put NOTE: 24Sep2014  JR         7) Amended Gum to gum in headers;
%put NOTE: ;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

*=====;
* START OF PROGRAM CODE                               ;
*=====;

/* Standard - just change the number to match the listing you're working
on. Also change the letters in the*/
/* bracket, eg ccb = current cigarette brands. Make sure to do this at
the top of the code too. */
%let tflno=T_15_02_06_16(bmi);

```

```

/* Standard - leave this */
%let TFL_Part=%scan(&_SASPROGRAMFILE,-3,%str(/));

/* Standard - leave this */
data _null_;
    tmp="%TFL_Part";
    if tmp not in ("dev" "qc") then call symput("TFL_Part", "prod");
    call symput('TFLpath', compress("&_SASPROGRAMFILE",""));
run;

*****;
* read in data ;
*****;
/*Obtaining treatments*/
data dummytrts;
    attrib trtseqa length=$200.
           trtseqan length=8.;

    trtseqa="THS 2.2 Menthol - mCC";
    trtseqan=1;
    output;
    trtseqa="mCC - THS 2.2 Menthol";
    trtseqan=2;
    output;
    trtseqa="THS 2.2 Menthol - NRT gum";
    trtseqan=3;
    output;
    trtseqa="NRT gum - THS 2.2 Menthol";
    trtseqan=4;
    output;
    trtseqa="Enrolled not randomized";
    trtseqan=5;
    output;
    trtseqa="Overall Safety";
    trtseqan=99;
    output;

run;
proc sort data=adam.adsl(where = (saf1 = 'Y' and enr1 = 'Y'))
out=adsl;
    by subjidn;
run;

data adsl2a;
    set adsl;
    if index(trtseqa,'Exposed') or missing(trtseqa) then delete;
    output;
    trtseqa='Overall Safety';
    trtseqan=99;
    output;
    keep subjidn trtseqa trtseqan ;
run;

proc freq data=adsl2a;

```

```

        table trtseqa*trtseqa/ noprint out=trt(drop=percent);
run;
proc sort data=trt;
    by trtseqa;
run;

data trt1;
    merge dummytrts(in=b) trt(in=a);
    by trtseqa;
    if a or b;
    if b and not a then count=0;
    call symput('trt' || compress(put(trtseqa,best.)),
compress(count));
run;

/*Start of code for BMI classifications*/

data adsl_bmi1;
    set /*adsl*/ADAM.ADVS; /* 6) JMH 23Sep2014 */
    WHERE PARAMCD='DBMI'; /* 6) JMH 23Sep2014 */
    IF ABLFL='Y' THEN AVISIT='Baseline'; /* 6) JMH 23Sep2014 */
    if index(trtseqa,'Exposed') or missing(trtseqa)then delete;
    output;
    trtseqa='Overall Safety';
    trtseqa=99;
    output;
run;

proc sort data=adsl_bmi1;
    by trtseqa trtseqa /*bmigrln bmigr1*/AVISITN AVISIT AVALCAT1; /*
6) JMH 23Sep2014 */
run;

proc sort data=adsl_bmi1 nodupkey out=bmi3b(keep=trtseqa trtseqa AVISITN
AVISIT); /* 6) JMH 23Sep2014 */
    by trtseqa trtseqa AVISITN AVISIT; /* 6) JMH 23Sep2014 */
run;

data bmi3b_x1;
    set bmi3b;
    /*bmigr1*/AVALCAT1 = 'Underweight'; /* 6) JMH 23Sep2014 */
/*    bmigrln=1;*/
/*    dumflag=1;*/
/*    count=0;*/
run;
/* 6) START JMH 23Sep2014 */

DATA BMI3B_X2;
    SET BMI3B;
    AVALCAT1 = 'Obese';
RUN;

DATA BMI3B_X;
    SET BMI3B_X1 BMI3B_X2;

```

```

RUN;

PROC SORT DATA=BMI3B_X; BY TRTSEQAN TRTSEQA AVISITN AVISIT AVALCAT1; RUN;
/* 6) END JMH 23Sep2014 */

data adsl_bmi;
    merge adsl_bmi1(in=a) bmi3b_x(in=b);
    by trtseqan trtseqa /*bmigr1n bmigr1*/AVISITN AVISIT AVALCAT1; /*
6) JMH 23Sep2014 */
    if a or b;
    IF B AND NOT A THEN DUMFLAG=1; /* 6) JMH 23Sep2014 */
run;

data trtl_a;
    set trtl;
    rename count = total;
run;

proc freq data=adsl_bmi noprint;
    table
    trtseqan*trtseqa*AVISITN*AVISIT*AVALCAT1/*bmigr1n*bmigr1*/dumflag / out
    =bmi1(drop=percent); /* 6) JMH 23Sep2014 */
run;

data bmi2;
    merge bmi1(in=a) trtl_a;
    by trtseqan trtseqa;
    if a;
    percent=count/total*100; /*This works out the percentages*/

run;

data bmi3;
    set bmi2;
    attrib avalc avalc2 format=$20.
        stat percentc statval statval2 format=$100.;

    BMIGR1=AVALCAT1; /* 6) JMH 23Sep2014 */

    paramn=16;
    paramcd='BMI (kg/m${super 2})';
/*    avisitn=99; */ /* 6) JMH 23Sep2014 */
/*    avisit='Day -1'; */ /* 6) JMH 23Sep2014 */

    if bmigr1='Underweight' then do; stat='Underweight
(<18.5kg/m${super 2}) - n (%)'; order2=2000; end;
    else if bmigr1='Normal weight' then do; stat='Normal range (>=18.5
and <25kg/m${super 2}) - n (%)'; order2=2001; end;
    else if bmigr1='Overweight' then do; stat='Overweight (>=25 and
<30kg/m${super 2}) - n (%)'; order2=2002; end;

```

```

        else if bmigr1='Obese' then do; stat='Obese ( $\geq 30\text{kg/m}^2$ ) -
n (%)'/*'Obese ( $\geq 25$  and  $< 30\text{kg/m}^2$ ) - n (%)'*/; order2=2003;
end;/* 1) AOB 12Aug2014 */
        else if missing(bmigr1) then do; stat='Missing n (%)'; order2=2004;
end;

```

```

        if not missing(bmigr1) then stat=left(trim(bmigr1)) || ' - n (%)';

        countc=left(compress(put(count,8.)));

        if count ge 10 and count lt 100 then count1=compress(countc);
        else if count lt 10 then count1=' '||compress(countc);

        percentc=left(compress(put(round(percent,0.1),5.1))) || '%';

        if percent=100 then percent1='(100 %)';
        else if percent ge 10 then percent1='( ' ||
compress(percentc) ||')';
        else if percent lt 10 then percent1='( ' ||
compress(percentc) ||')';

        statval=trim(count1);
        statval2=percent1;

```

```

        if dumflag ne 1 and order2 not in(2000 2001 2002 2003 2004) then
do;
            avalc = strip(put(count,best.));
            avalc2='(' ||strip(put(round(percent,0.1),5.1)) || ')';
        end;

        if dumflag ne 1 and order2 in (2000 2001 2002 2003 2004) then do;
            avalc = compress(count1);
            avalc2=percent1;
        end;

        else avalc='0';
run;

```

```

proc sort data=bmi3;
    by paramn paramcd avisitn avisit order2 stat;
run;

```

```

proc transpose data=bmi3 out=bmi4 prefix=t;
    by paramn paramcd avisitn avisit order2 stat;
    id trtseqan;
    idlabel trtsega;
    var avalc;
run;

```

```

proc transpose data=bmi3 out=bmi4p prefix=p;
    by paramn paramcd avisitn avisit order2 stat;
    id trtseqan;

```

```

        idlabel trtseqa;
        var avalc2;
run;

data bmi4all;
    merge bmi4 bmi4p;
    by paramn paramcd avisitn avisit order2 stat;
    drop _name_;
run;

/*End of BMI classes. Set on with rest of stats later*/

/* All other data */
proc sort data=adam.advs(where = (saffl = 'Y' and enrfl = 'Y' AND
ANL01FL='Y' and paramcd in('BMI' 'WEIGHT' 'HEIGHT' 'DBMI'))) out=advs;
    by subjidn paramcd avisit;
run;

/* These are all the variables required from ADVS */
data advs2_a;
    set advs;
    if index(trtseqa,'Exposed') or missing(trtseqa)then delete;
    if paramn ne 22 then do;
        paramcd1 = left(trim(paramcd))||'
'||left(compress(avalu))||')';
    end;
    else do;
        paramn=/*16*/15.5; /* 6) JMH 23Sep2014 */
        paramcd1 = left(trim(paramcd))||'${super 1}
'||left(compress(avalu))||')';
        paramcd1=tranwrd(paramcd1,'DBMI','BMI');
    end;
    output;
    trtseqa='Overall Safety';
    trtseqan=99;
    output;

        drop paramcd;
run;

data advs_orig;
    set advs2_a;
run;

data advs_test;
    set advs2_a;
    drop aval avisit;
run;

data advs_chg;
    set advs_test(where=(avisitn in(104)));
    attrib avisit length=$40.;
    if avisitn=104 then avisitn=401;/*change from baeline to
discharge*/

```

```

        aval=chg;
        avisit='Change from Day -1 at Day 4';
run;

data advs2;
    set advs_orig advs_chg;
    paramcd=paramcd1;
    if paramcd='HEIGHT (cm)' then do;
        aval=aval/100;
        avalc=left(trim(put(aval,best.)));
        avalu='m';
        paramcd='HEIGHT (m)';
    end;

    keep usubjid paramcd param paramn trtsega trtseqan avisit avisitn avalc
    aval avalu;
run;

proc sort data=advs2;
    by usubjid;
run;

data advs3;
    merge advs2(in=a) adsl;
    by usubjid;
    if a;
    keep trtseqan trtsega paramn paramcd param avisitn avisit avalc
    aval avalu ;
run;

proc sort data=advs3;
    by trtseqan trtsega paramn paramcd param avisitn avisit avalc aval
    avalu;
run;

%macro Stats(visit=, set=, stats=);
    proc means data=advs3 noprint;
        by trtseqan trtsega paramn paramcd param avisitn avisit;
        var aval;
        where avisit=&visit;
        output out=&stats;
        output out=&set median=;
    run;

%mend stats;

%stats(visit='Screening', set=scrn, stats=st1);
%stats(visit='Day -1', set=admi, stats=st2);
%stats(visit='Day 4/Discharge', set=disc, stats=st3);
%stats(visit='Change from Day -1 at Day 4', set=chg, stats=st4);

data all;

```

```

set scrn st1 admi st2 disc st3 chg st4;

if missing(_stat_) then _stat_ = 'Median';
if _stat_='N' then order2=1001;
if _stat_='MEAN' then order2=1002;
if _stat_='STD' then order2=1003;
if _stat_='Median' then order2=1004;
if _stat_='MIN' then order2=1005;
if _stat_='MAX' then order2=1006;

attrib stat format=$100.
      avalc length=$20.;
if _stat_ eq 'STD' then stat='SD';
else stat=propcase(_stat_);
drop _stat_;

if stat='N' then avalc=left(trim(put((round(aval,1)),8.))),);

ELSE if paramcd ne 'HEIGHT (m)' then do;/* 3) AOB 12Aug2014 */
      if stat='SD' then
avalc=compress(put(0.001*ceil(aval/0.001),8.3));
      else if stat in('Mean' 'Median') then
avalc=left(trim(put((round(aval,0.01)),8.2))),);
      else avalc=left(trim(put((round(aval,0.01)),8.1))),);
end;
else do;
      if stat='SD' then
avalc=compress(put(0.0001*ceil(aval/0.0001),8.4));
      else if stat in('Mean' 'Median') then
avalc=left(trim(put((round(aval,0.001)),8.3))),);
      else avalc=left(trim(put((round(aval,0.001)),8.2))),);
end;

run;

data meansd;
      set all (keep=trtsega trtsega stat avalc paramcd paramn order2
avisitn avisit);
      where stat in('Mean' 'SD'); /*Only keep mean and SD as this is all
we want here*/
run;

proc sort data=meansd;
      by trtsega trtsega paramn paramcd avisitn avisit;
run;

proc transpose data=meansd out=meansd1 prefix=m;
      by trtsega trtsega paramn paramcd avisitn avisit;
      id order2;
      idlabel stat;
      var avalc;
run;

data meansd2;

```



```

        format stat avalc $20.;
        set meansd1;
        if not missing(m1002) and not missing(m1003) then do;
            avalc=left(compress(m1002)) || ' (' || left(compress(m1003))
|| ')';
        end;
        else if not missing(m1002) and missing(m1003) then do;
            avalc=left(compress(m1002));
        end;
        else if missing(m1002) and missing(m1003) then do;
            avalc='';
        end;
        stat='Mean (SD)';
        order2=1002;
        drop m1002 m1003;
run;

        /*End of combining mean and SD*/

/*Combine Min and Max*/
data minmax;
    set all(keep=trtseqan trtseqa stat order2 avalc paramn paramcd
avisitn avisit);
    where stat in('Min' 'Max'); /*Only keep min and max as this is all
we want here*/
run;

proc sort data=minmax;
    by trtseqan trtseqa paramn paramcd avisitn avisit;
run;

proc transpose data=minmax out=minmax1 prefix=m;
    by trtseqan trtseqa paramn paramcd avisitn avisit;
    id order2;
    idlabel stat;
    var avalc;
run;

data minmax2;
    format stat avalc $20.;
    set minmax1;
    if not missing(m1005) and not missing(m1006) then do;
        avalc=left(compress(m1005)) || ', ' || left(compress(m1006));
    end;
    else if missing(m1005) and missing(m1006) then do;
        avalc='';
    end;
    stat='Min, Max';
    order2=1005;
    drop m1005 m1006;
run;

/*End of combining min and max*/

/*Combine the new datasets which have Min, Max and Mean (SD)*/
data minmaxmeansd;

```

```

        set minmax2 meansd2;
run;

data all2;
    set all(where=(order2 ne 1002 and order2 ne 1003 and order2 ne 1005
and order2 ne 1006)) minmaxmeansd;
run;

data all3;
    set all2;
    if paramn=14 then paramn=17;
run;

proc sort data=all3;
    by trtseqan trtseqa;
run;

data all4;
    merge all3(in=a) dummytrts(in=b);
    by trtseqan trtseqa;
    if a or b;

    if b and not a then do;
        avalc = '';
        paramn=15;
        param='Weight';
        paramcd='WEIGHT (kg)';
        avisitn=1;
        avisit='Screening';
        order2=1001;
        stat='N';
    end;

    if paramn=16 AND PARAMCD NE 'BMI${super 1} (kg/m2)' then
paramcd='BMI${super 1} (kg/m${super 2})';
    else if paramn=16 and paramcd='BMI${super 1} (kg/m2)' then
paramcd='BMI${super 1} (kg/m${super 2})';
run;

proc sort data=all4;
    by paramn paramcd avisitn avisit order2 stat ;
run;

proc transpose data= all4 prefix=t out=all5;
    by paramn paramcd avisitn avisit order2 stat;
    var avalc;
    id trtseqan;
    idlabel trtseqa;
run;

data all6;
    set all5 bmi4ALL;
/* START 2) AOB 12Aug2014 */
    ATTRIB      T1 LABEL= "THS 2.2 Menthol#- mCC #(N=&trt1)"

```

```

                T2 LABEL="mCC -#THS 2.2 Menthol#(N=&trt2)"
                T3 LABEL="THS 2.2 Menthol -#NRT gum #(N=&trt3)" /* 7) JR
24Sep2014 */
                T4 LABEL="NRT gum#- THS 2.2 Menthol#(N=&trt4)" /* 7) JR
24Sep2014 */
                T5 LABEL="Enrolled Not#Randomized#(N=&trt5)"
                T99 LABEL="Overall#Safety#(N=&trt99)";
/* END 2) AOB 12Aug2014 */
    if stat = 'N' or order2 >=2000 then do;
        if missing(t1) then t1='0';
        if missing(t2) then t2='0';
        if missing(t3) then t3='0';
        if missing(t4) then t4='0';
        if missing(t5) then t5='0';
    end;

    PARAMCD=TRANWRD(PARAMCD,'EIGHT','eight');
    if stat = 'N' then stat = 'n';

    flag=1;

    if index(avisit,'/') then avisit=tranwrd(avisit,'/',' ');
    if avisit='Day -1' then avisit='Baseline';
    if avisit='Change from Day -1 at Day 4' then avisit='Change from
Baseline';

/*    if paramn in(15 16) then do;*/ /* 6) JMH 23Sep2014 */
        if avisitn in(/*1*/ 401) then delete; /* 6) JMH 23Sep2014 */
/*    end;*/ /* 6) JMH 23Sep2014 */

        /*We only want Height at Screening*/
        IF AVISITN=1 AND PARAMN NE 17 THEN DELETE; /* 6) JMH
23Sep2014 */
run;

proc sql noprint;
    create table table.T_15_02_06_16 as
    select paramcd, avisit, stat, t1, P1, t2, P2, t3, P3, t4, P4, t5,
t99, P99
    from all6
    order by paramn, avisitn, order2;
quit;

proc sort data=all6;
    by paramn avisitn order2;
run;

data paging;
    set all6;
    by paramn avisitn order2;
/*    if (paramn=16 and order2=2000) or (paramn=16 and avisitn=104
and order2=1001) then ln=1;*/
    IF FIRST.PARAMN THEN LN=1; /* 6) JMH 23Sep2014 */
    else ln+1;

```

```

        if ln=1 then page+1;
        call symput("page",compress(put(page,best.)));
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsz missing=
';
ods escapechar='$';
%let linetop = \brdrt\brdrs\brdrw30; * needs to be 1.5pt so calculated
in twips (1/20 pt) ;
%let linebot = \brdrb\brdrs\brdrw30;
%let linebot2 = \brdrb\brdrs\brdrw15;

/* Standard - macro for paging */
%macro outrtf(blankn=, halfblnk=);

%if &halfblnk=N %then %let halfblnk=;
%else %if &halfblnk=Y %then %let halfblnk=\~;

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106326/TFL/&TFL_Part./&tflno..rtf"
style=tl06326 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;
ods proclabel = ' ';

data comp;
    set paging end=eof;
    by paramn avisitn order2;
    where page=&i;

    /* Amend title as needed */
    _firtitl="Table 15.2.6.16 Summary of Weight and BMI Measurements -
Safety Population";
    _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(Page &i of &page)");
    if eof then do;
        call symput('_FSRTITL', trim(left(_firtitl)));
        call symput('_blankn', compress(put(len,best.)));
    end;
    call symput('ord',order2);
    drop _firtitl _upcas len;
run;

```

```

* most set up in template others below;
* title arial 12pt bold with 12pt paragraph space below;
* all headers to be arial 11pt bold;
* data arial 10pt;
* headers to be central, text values left aligned and numeric centered
around decimal point;
/* Update with your variables as needed */
proc report data = comp headline headskip missing nowd split = '#' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
    column flag page
%IF &ORD LT 2000 %THEN %DO; paramn paramcd avisitn avisit order2 stat
                                ("Sequence &linebot"
t1 t2 t3 t4 t5) t99 %end;
%else %do; paramn ("Parameter (units)" paramcd) avisitn ("Study Day"
avisit) order2 ("Statistic" stat)
("Sequence &linebot" ("THS 2.2 Menthol#- mCC #(N=&trt1)"/*"THS 2.2
Menthol -#mCC #(N=&trt1)"/ t1 p1) ("mCC -#THS 2.2 Menthol#(N=&trt2)" t2
p2) /* 2) AOB 12Aug2014 */
("THS 2.2 Menthol -#NRT gum #(N=&trt3)" t3 p3) ("NRT gum#- THS 2.2
Menthol#(N=&trt4)"/*"NRT Gum -#THS 2.2 Menthol#(N=&trt4)"/ t4 p4) /* 2)
AOB 12Aug2014 */
("Enrolled Not#Randomized#(N=&trt5)" t5 p5)) ("Overall#Safety#(N=&trt99)"
t99 p99) %end;;
    define flag                / order order = internal noprint;
    define page                / order order = internal noprint;
    define paramn              / order order=internal noprint;
    %if &ord lt 2000 %then %do;
        define paramcd        / order style={just=left cellwidth=3.0cm}
STYLE(HEADER)={JUST=CENTER} "Parameter (units)";
        define avisitn        / order order=internal noprint;
        define avisit         / group style={just=left cellwidth=2cm}
STYLE(HEADER)={JUST=CENTER} "Study Day";
        define order2         / order order=internal noprint;
        define stat           / display style={just=left
cellwidth=2cm} STYLE(HEADER)={JUST=CENTER} "Statistic";
        define t1-t5          / display style={just=c
cellwidth=2.2cm} style(header)={just=center};
        define t99            / display style={just=c
cellwidth=2.2cm} style(header)={just=center};
    %end;
    %else %do;
        define paramcd        / order style={just=left
cellwidth=/*3.0*/2.8cm} style(header)={just=center} ""; /* 4) AOB
12Aug2014 */
        define avisitn        / order order=internal noprint;
        define avisit         / group style={just=left
cellwidth=/*2.1*/1.8cm} style(header)={just=center} ""; /* 4) AOB
12Aug2014 */ /* 4) AOB 12Aug2014 */
        define order2         / order order=internal noprint;
        define stat           / display style={just=left
cellwidth=3cm} style(header)={just=center} "";
        define t1-t5          / display style={just=c cellwidth=0.5cm}
style(header)={just=center} "";

```

```

        define t99                / display style={just=c
cellwidth=0.5cm} style(header)={just=center}"";
        define p1                / display style={just=c
cellwidth=/*1.2*/1.35cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */
        define p2                / display style={just=c
cellwidth=/*1.2*/1.35cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */
        define p3                / display style={just=c
cellwidth=/*1.2*/1.35cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */
        define p4                / display style={just=c
cellwidth=/*1.2*/1.35cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */
        define p5                / display style={just=c
cellwidth=/*1.4*//*1.35*/1.5cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */ /* 5) JMH 12Aug2014 */
        define p99              / display style={just=c
cellwidth=/*1.2*/1.35cm} style(header)={just=center}""; /* 4)  AOB
12Aug2014 */
        %end;

break before flag / page %if &i=1 %then %do;
contents="&_fsrtitl" %end; %else %do; contents='' %end;;

break after page / page;

compute before page / style={protectspecialchars=off};
    line "&linetop";
endcomp;

compute before _page_ / style={just=left protectspecialchars=off};
    line "\b\fs24\sa24&_FSRTITL." ; * \b = bold, \fs24 is font
size 12pt, \sa24 is space after 12pt;
    line "&linebot" ;
endcomp;

compute after avisit;
    line " ";
endcomp;

compute after _page_ / style={just=left protectspecialchars=off
pretext="&linetop."};
    line 'Note: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';
    line 'Note: Enrolled Not Randomized refers to all subjects
enrolled but not randomized. Overall Safety refers to all subjects
exposed to THS 2.2 Menthol or NRT gum.';
    line "Note: Percentages are based on the number of subjects
indicated in the column header (N).";
/*    line "Note: Baseline is Day -1.";*/

```

```

        LINE "Note: Baseline is the last available time point prior
to the product test (THS 2.2 Menthol or NRT gum) at Admission (Day -1).";
/* 6) JMH 23Sep2014 */
        line "Note: 1: BMI is derived in analysis datasets.";
        line "";
        line "Appendix 15.3.6.10";
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of &page)";
        line "Program Run: &sysdate &sysuserid   Program Status:
&status";
        endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=70, halfblnk=N);
ods listing;
proc printto print = "&table./T_15_02_06_16.lst" new;
run;

proc contents data = table.T_15_02_06_16 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```